REMARKS

Claims 1-20 remain pending in the present application as amended. Independent claims 1, 8, and 14 have been amended. No claims have been added or canceled. Applicants respectfully submit that no new matter has been added to the application by the Amendment. In particular, Applicants respectfully submit that the added language in the claims regarding the user and kernel modes is disclosed in the published application at least in connection with Figure 4.

Telephone Conversation With Examiner

Examiner Joseph is thanked for the telephone conversation conducted on February 24, 2010. Proposed amendments were discussed. Asserted art was discussed. It appears that the proposed amendments overcome the rejections based on the asserted art.

Claim Rejection

The Examiner has again rejected the claims under 35 U.S.C. § 103 as being obvious over admitted prior art (APA) of the present application as disclosed in connection with Fig. 3, and further in view of Price et al. (U.S. Patent Pub. No. 2002/0120607) and Vermeire et al. (U.S. Patent Pub. No. 2001/0025372). Applicants respectfully traverse the Section 103 rejection insofar as it may be applied to the claims as amended. In particular, Applicants respectfully submit that per the claims as amended, the pen services is in a user mode, which is contrary to the reading of the admitted prior art as asserted by the Examiner.

To remind the Examiner, in the present application, pen data from a digitizer of a computing device or information derived from such pen data is employed by an application on the computing device, where the application includes managed code. The pen data relates to movement of a stylus with respect to the digitizer, and thus includes information relating to one or more locations of the stylus with respect to the digitizer. Although the application employs managed code to process the pen data, the computing device employs a non-application component (such as a pen service) to initially receive the incoming pen data, where the component is not managed code but instead is written as unmanaged code. Typically, although not necessarily, the pen service is provided by the operating system of the computing device while the application is not likewise provided by such operating system.

To also remind the Examiner, managed code and unmanaged code relate to use of a framework on the computing device such as the .NET framework supplied by MICROSOFT Corporation of Redmond, Washington. Such .NET framework in particular includes a large library of pre-coded solutions to common programming problems, a runtime or virtual machine that manages the execution of programs written specifically for the framework, and a set of tools for configuring and building applications. Programs written for the .NET Framework (i.e., 'managed code') execute in a software environment that manages the program's runtime requirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine so that programmers need not consider the capabilities of the specific processor that will execute the program. As a result, managed code is executed under the CLR in a manner independent of the processor of the computing device. In contrast, 'unmanaged code' is not written for such framework and is not executed under the CLR. Instead, such unmanaged code is native to and executed directly by the processor of the computing device.

To further remind the Examiner, interaction between managed and unmanaged code is commonly required, such as for example to transfer pen data from a digitizer to a .NET application. Accordingly, a framework such as the .NET Framework provides a mechanism to access functionality that is implemented in programs that execute outside the .NET environment. For example, and as set forth in the application at paragraph [0009] (as published), a callable run-time wrapper as shown in Fig. 3 may be invoked to retrieve information from a native component, where every call from managed space to the native component passes through standard interop data marshalling (data transfer) as established by the runtime callable wrapper. However, and significantly, such standard interop data marshalling (data transfer) as employed in the .NET framework, for example, extends across a metaphorical border between managed and unmanaged code and therefore is relatively slow, which can be an impediment to high data throughput. Moreover, the wrapper is *external* to an application (as is shown in Fig. 3) and therefore represents an additional layer through which data must pass prior to reaching the application.

Accordingly, in the present application and as shown in Fig. 4, data transfer is instead achieved by way of shared memory and pointers as well as by employing entities *of the application* that do not extend across the aforementioned metaphorical border between managed and unmanaged code, as is shown in Fig. 4. By doing so, packet transfer speed for pen data information may be increased by a factor of 2-5 times as compared with the standard CLR interop scheme (paragraph [0055]).

In asserting the present Section 103 rejection, the Examiner has chosen to interpret several elements shown in the Fig. 3 as reading on the subject matter shown in Fig. 4, and in particular has asserted that the recited pen service of independent claims 1, 8, and 14 is shown in Fig. 3 as the APA component [Pen Device Drivers] 301 (*See* Office Action at 3, 4). As seen in Fig. 3, the component 301 operates in the *kernel mode* of the computing device. However, the pen service [402] of the present application as shown in Fig. 4 is in fact operating in the *user* mode of the computing device. Accordingly, Applicants have amended the independent claims to emphasize this distinction. In particular, Applicants respectfully point out that in the claims as amended, the pen service is recited as *operating in a user mode on the computing device* and receives pen data from *a pen device driver operating in a kernel mode on the computing device*. Accordingly, Applicants respectfully submit that the Examiner's interpretation of the elements of Fig. 3 can not be said to read upon the subject matter now recited in the independent claims as amended.

In setting forth the present Section 103 rejection, the Examiner has also asserted that the recited pen input component of independent claims 1, 8, and 14 is shown in Fig. 3 as the APA component 302, and that the recited stylus input subsystem of independent claims 1, 8, and 14 is shown in Fig. 3 as the APA component 305 (*See* Office Action at 4). However, Applicants respectfully point out that both components 302 and 305 are shown in Fig. 3 as being *exterior* to the application 307 of such Fig. 3 and therefore are not constituent parts of the application 307 of

Fig. 3. Accordingly, Applicants respectfully submit that the non-application components 302 and 305 can not be employed to read upon the pen input component and stylus input subsystem of independent claims 1, 8, and 14, especially inasmuch as such pen input component is recited as being *loaded into* the recited application and such stylus input subsystem is recited as being *of* the recited application, and accordingly are constituent elements of the recited application. As a result, Applicants respectfully submit that the aforementioned elements of Fig. 3 can not be said to read upon the subject matter recited in the independent claims in this regard either.

Applicants respectfully note here that the subject matter shown in Fig. 3 differs from the subject matter recited in claims 1, 8, and 14 in large part because the subject matter of Fig. 3 (rather clumsily) crosses the dividing line between managed and unmanaged code. In doing so, interop data marshalling is employed to pass pen data, which as has already been pointed out is relatively slow and can represent a hindrance to effective pen data processing as the amount of pen data increases. In contrast, the subject matter recited in claims 1, 8, and 14 more elegantly respects the aforementioned dividing line between managed and unmanaged code. Thus, such claimed subject matter does not employ interop data marshalling and instead employs faster shared memory. Quite simply, the subject matter shown in Fig. 3 expresses no appreciation for the need to increase the speed of passing pen data to compensate for increasing amounts of pen data, and correspondingly expresses no appreciation that such increased speed should or could be achieved through the use of shared memory and code that does not cross the dividing line between managed and unmanaged code.

Applicants respectfully point out that independent claims 1, 8, and 14 as amended all recite a particular transfer routing of a pointer. Specifically, such claims require that (A) a pen service receives the pointer from a shared memory, (B) the pen service transfers the received pointer to a loaded pen input component of the application, (C) the loaded pen input component further transfers the received pointer to a stylus input subsystem of the application, and (D) the stylus input subsystem submits the received pointer with a retrieval command to the shared memory *by way of an exposed P-Invoke-style interface to the pen input component* to retrieve information from said shared memory by way of the transferred pointer. Applicants note that the

Examiner cites to the Vermeire reference as disclosing the user of pointers in general. However, Applicants respectfully submit that neither the alleged APA of Fig. 3 nor the Vermeire reference discloses or even suggests the specific use of a pointer as is recited in independent claims 1, 8, and 14, and in particular neither reference discloses or even suggests that such a pointer should or could be passed among specific elements in the manner that is positively recited in the independent claims. Accordingly, Applicants respectfully submit that the alleged APA of Fig. 3 and the Vermeire reference can not be said to read upon the subject matter recited in the independent claims in this regard too.

Applicants note here that the Price reference is cited only for the purpose of disclosing the use of a shared memory, and the Vermeire reference is cited only for the purpose of disclosing the use of pointers. Accordingly, Applicants respectfully submit that the Price and Vermeire references are otherwise inapposite to the subject matter now recited in claims 1, 8, and 14.

Thus, Applicants respectfully submit that the alleged APA in combination with the cited Price and Vermeire references fails to disclose or even suggest all of the above-mentioned elements as are now recited in independent claims 1, 8, and 14 of the present application as amended. Therefore, Applicants respectfully submit that the alleged APA and such references can not be combined to make obvious the subject matter recited in the independent claims as amended or the claims depending therefrom. Accordingly, Applicants respectfully request reconsideration and withdrawal of the Section 103 rejection.

CONCLUSION

In view of the foregoing Amendment and Remarks, Applicants respectfully submit that the present application including claims 1-20 is in condition for allowance and such action is respectfully requested.

Respectfully Submitted,

Date: April 9, 2010

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439